CSE 303, Winter 2009, Assignment 1B
Due: Thursday, January 15 at 11 pm

In this assignment you'll gain experience writing very short scripts.

General hint: It can be helpful to experiment with commands interactively in a shell window before you embed them in a script.

1. (An alias) Create a bash alias `private` such that when you run `private foo`, the entire subtree of the file-system starting at `foo` (so just `foo` if it is a file, but `foo` and all its files and subdirectories recursively if it is a directory) has its permissions changed as follows:

   - The user's permissions are unchanged.
   - All group and world permissions are set to no access of any sort.

   Put your alias in a file `defprivate` such that entering `source defprivate` would make `private` available in the shell.

2. (Script) Create a `bash` script `combine` that takes 2 or more arguments, call them $f1$, $f2$, …, fn. Script `combine` should work as follows:

   - All arguments are treated as filenames.
   - If fewer than two arguments are given, print a suitable error message on `stderr` and exit.
   - If a file or directory $f1$ already exists, print "Error: first file exists" on `stderr` and exit.
   - Otherwise concatenate the contents of $f2$, …, fn and copy them to `stdout`. Do not print any error messages from this (for example if some file does not exist or is a directory). Instead, any such error messages should be written to $f1$.

   Hint: Put filenames in double-quotes in case they contain "funny characters" (such as spaces). Your script should work with any file names, no matter what they contain.

   Hints: `shift`, `$@`, `-lt`, `-a`.

3. (Script) Create a bash script called `datedlinecount` that works as follows:

   - If it is given fewer than two arguments, it prints an appropriate error and exits.
   - Assume all the arguments are filenames for text files; you do not need to check for this.
   - Append to the file indicated by the first argument the following information:
     - The time and date
     - One line for each of the second-through-last arguments, containing the number of lines in the file and then the name of the file
     - If there were three or more arguments (i.e., two or more files to be counted), one additional line with the total number of lines in all the files and then the word "total"

   For example, executing: `./datedlinecount log foo bar; ./datedlinecount log foo*; cat log` might produce something like:

   ```
   Mon Mar 26 20:42:16 PDT 2007
     4 foo
    17 bar
    21 total
   Mon Mar 26 20:42:17 PDT 2007
    4 foo
    3 food
    7 total
   ```

Hints: `shift`, `date`, `wc`, `$@`.

Extra credit: The output should include a total line even if only a single file appears in the list.

Assessment: Your solutions should be:

- Correct scripts, etc. that run on `attu.cs.washington.edu`
- In good style, including indentation and line breaks.
- Of reasonable size

Identifying information including your name, CSE303 Homework 1B, the problem number, and the date should appear as comments in each of your files.

Turn-in Instructions Use the turnin dropbox linked on the course web page to submit your files. If you wish, you can combine your files into an archive (see the `tar` command) and turn that in as a single file instead of turning in the files individually. The choice is yours — do whatever is most convenient.